

HON 296

Prototyping with Arduino



open source
hardware



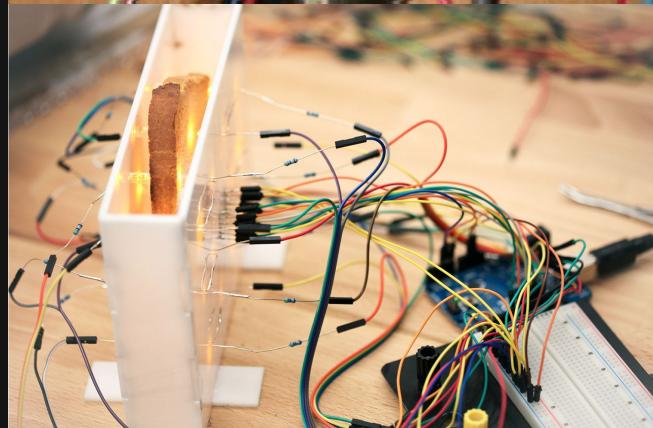
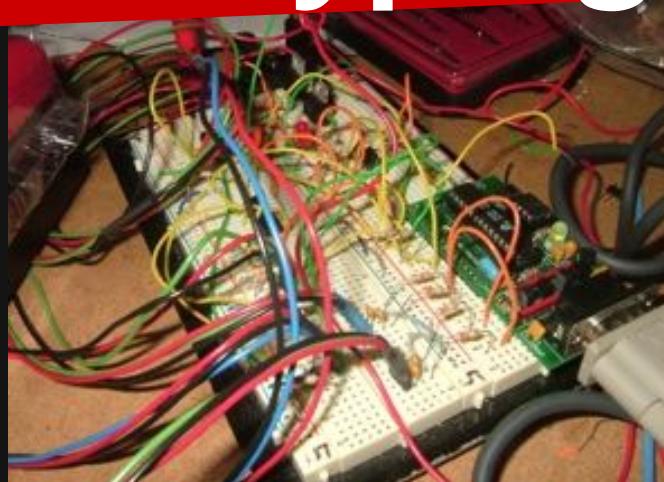
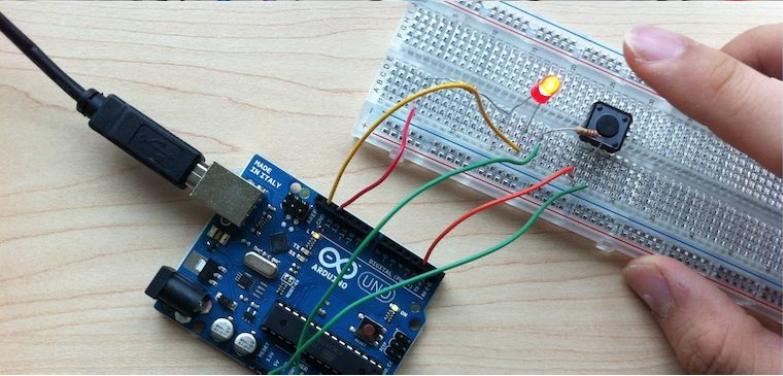
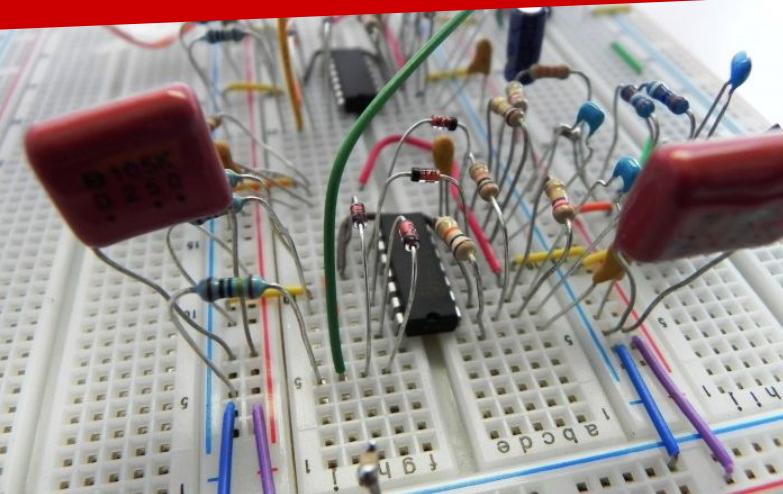
#WorkshopGoals

1. Show the potential of electronics and the Arduino platform.
2. Build your first *working* electronics project.
3. Build a Ghost detector
4. Offer resources for building more

Prototype

- To test a concept or process;
- A thing to be replicated or learned from.

Electronics Prototyping



Electronics Options

The library offers these options:



Sparkfun Arduino Inventor Kit w/ Wifi Shield

- Arduino Compatible
- Easiest to get started



Intel Galileo

- Arduino compatibility on top of embedded Linux.
- Wifi + ethernet
- More advanced and more powerful.



Raspberry Pi

- Full Linux with HDMI output
- Connect via Wifi or internet



LightBlue Bean

- Bluetooth-connected Arduino
- Tiny and low power
- Great for wearables and mobile applications

Arduino

ESP8266_Simple_Button_Light | Arduino 1.6.8

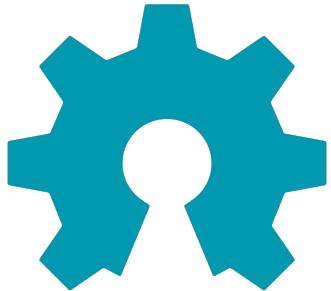
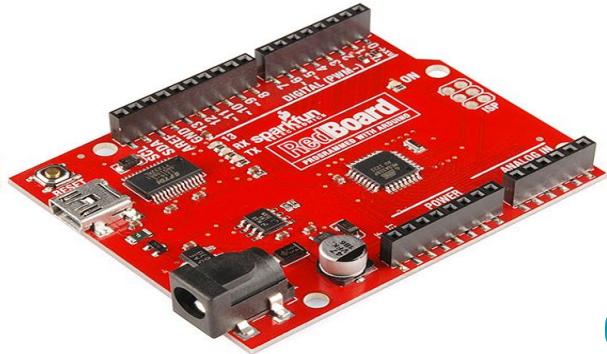
```
// constants won't change. They're used here to set pin numbers:  
const int buttonPin = 10; // the number of the pushbutton pin; note pin 10 here is pin SD3  
const int ledPin = 2; // the number of the LED pin; note pin 2 here is pin D4 on ESP8266  
  
// variables will change:  
int buttonState = 0; // variable for reading the pushbutton status  
  
void setup() {  
  // initialize the LED pin as an output:  
  pinMode(ledPin, OUTPUT);  
  // initialize the pushbutton pin as an input:  
  pinMode(buttonPin, INPUT);  
}  
  
void loop() {  
  // read the state of the pushbutton value:  
  buttonState = digitalRead(buttonPin);  
  
  // check if the pushbutton is pressed.  
  // if it is, the buttonState is HIGH:  
  if (buttonState == HIGH) {  
    // turn LED on:  
    digitalWrite(ledPin, HIGH);  
    delay(1000); // this will leave the light on for 1 second after the button is pressed  
  } else {  
    // turn LED off:  
    digitalWrite(ledPin, LOW);  
  }  
}  
  
Done uploading.  
at java.net.AbstractPlainDatagramSocketImpl.join(AbstractPlainDatagramSocketImpl.java:  
at java.net.MulticastSocket.joinGroup(MulticastSocket.java:323)  
at javax.jmdns.impl.JmDNSImpl.openMulticastSocket(JmDNSImpl.java:463)  
at javax.jmdns.impl.JmDNSImpl.<init>(JmDNSImpl.java:420)
```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (3M SPIFFS) on /dev/cu.wchusbserial1410

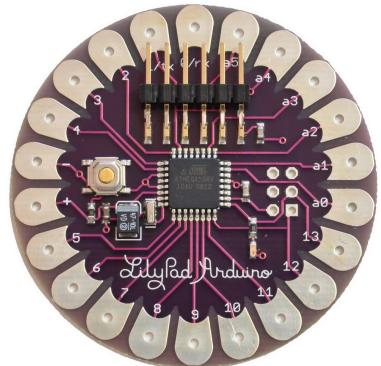


- *digital (code)*
+ *physical*
(electronics)
- inputs & outputs

(Lots of Arduinos)



open source
hardware



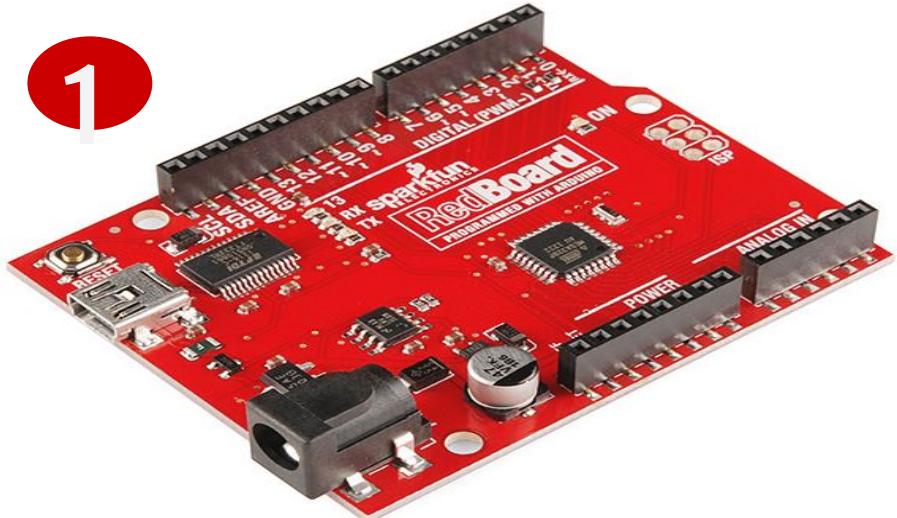
ARDUINO

An
Introduction
to
the
Arduino



Arduino

1



2

The screenshot shows the Arduino IDE interface. The title bar reads "StandardFirmata | Arduino 1.0.4". The main window displays the "StandardFirmata" tab and a sketch titled "helloworld". The code in the editor is:

```
/*
  Button

  Turns on and off a light emitting diode(LED) connected to digital
  pin 13, when pressing a pushbutton attached to pin 2.

  The circuit:
  * LED attached from pin 13 to ground
  * pushbutton attached to pin 2 from +5V
  * 10K resistor attached to pin 2 from ground

  * Note: on most Arduinos there is already an LED on the board
  attached to pin 13.

  created 2005
  by DojoDave <http://www.0j0.org>
  modified 30 Aug 2011
  by Tom Igoe

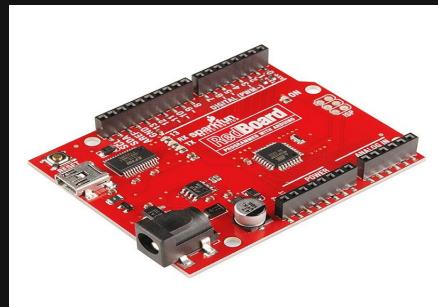
  This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Button
*/
// constants won't change. They're used here to
```

At the bottom of the screen, the status bar shows "56" and "Arduino Uno on /dev/cu.usbmodemfd121".

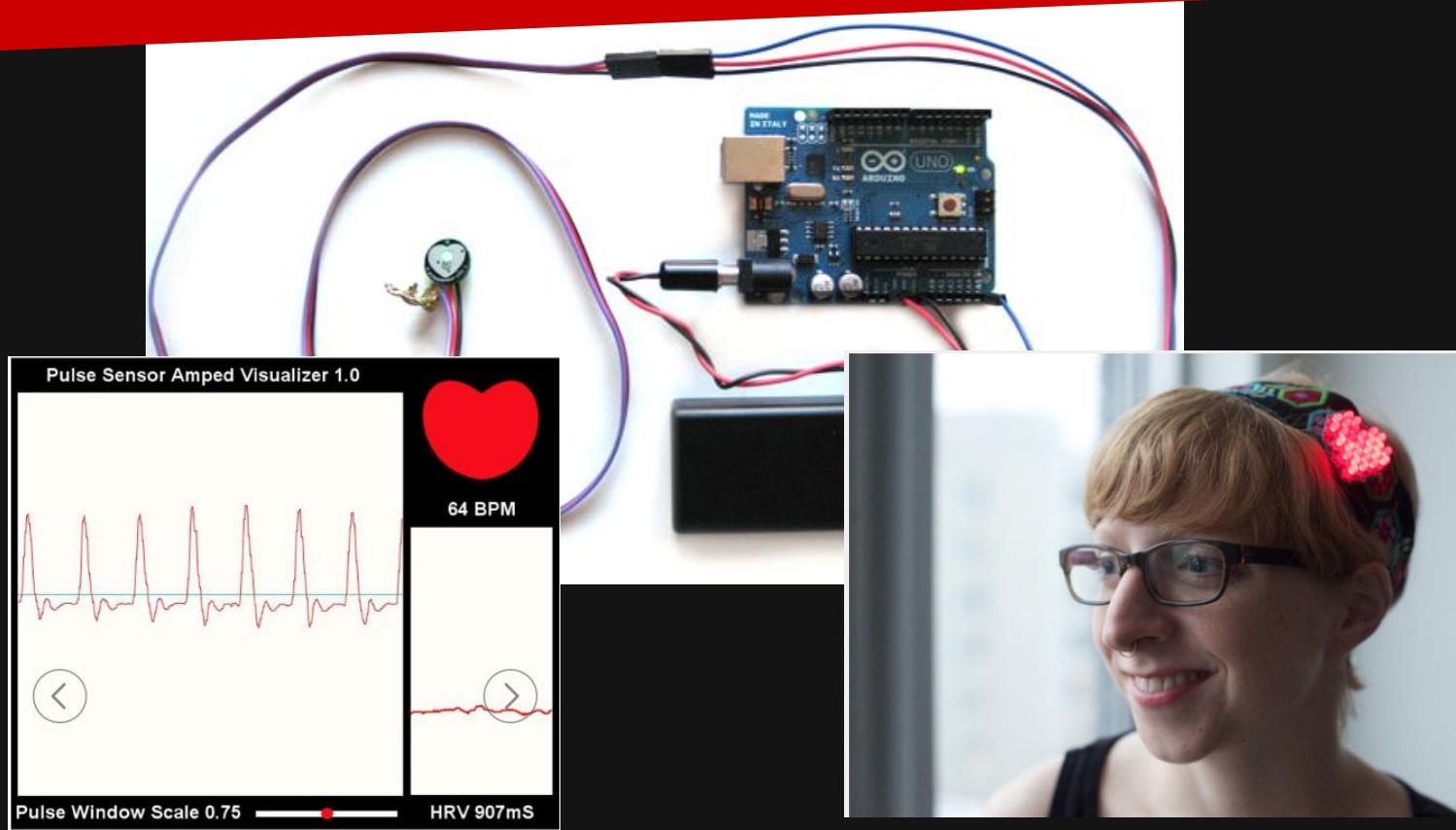
Inputs & Outputs

data streams
user interactions
moisture
human movement
text/multimedia
power/battery
light

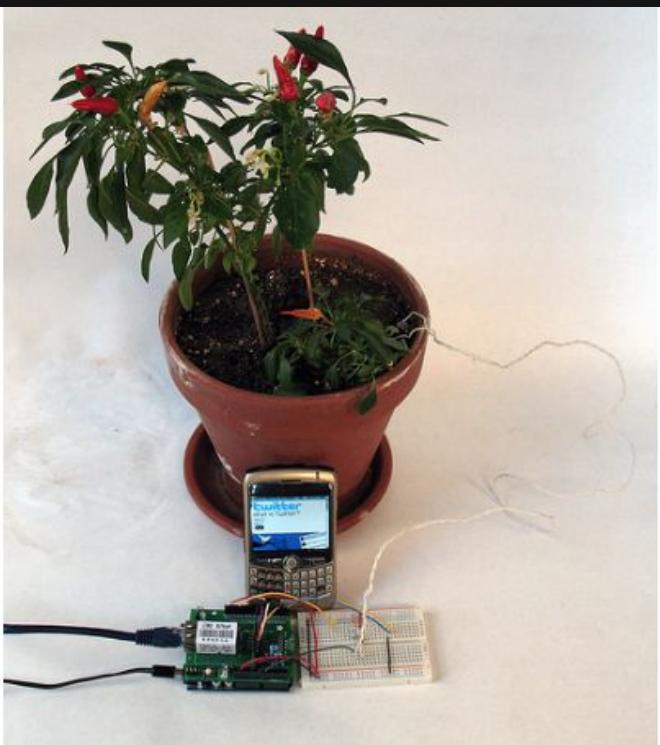


Twitter/web
data visualization
actuators
sound
light

Examples



Examples



Botanicalls @botanicalls · 29 Oct 2008
Thank you for watering me!

Botanicalls @botanicalls · 26 Oct 2008
URGENT! Water me!

Botanicalls @botanicalls · 24 Oct 2008
Water me please.

Botanicalls @botanicalls · 1 Oct 2008
You over watered me.

Botanicalls @botanicalls · 1 Oct 2008
Current Moisture: 64%.

Botanicalls @botanicalls · 11 Jun 2008
Water me please.

A screenshot of a Twitter feed showing five tweets from the account "Botanicalls". The tweets are timestamped from October 2008. The first three tweets are from 2008, and the last two are from 2006. The tweets are all variations of the plant asking for water or complaining about being overwatered. Each tweet includes a small profile picture of a person holding a plant, the account name "Botanicalls", the handle "@botanicalls", the date, and the text message. Below each tweet are standard Twitter interaction icons: a reply arrow, a retweet arrow, a star for favorites, and three dots for more options.

Your Tools

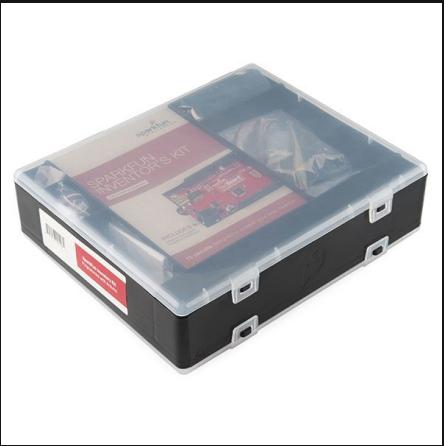
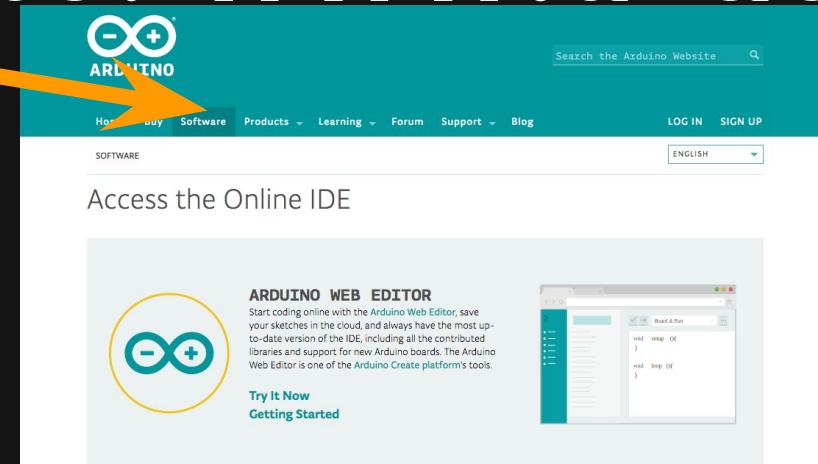


Photo: David Gerardo

Using your own computer?

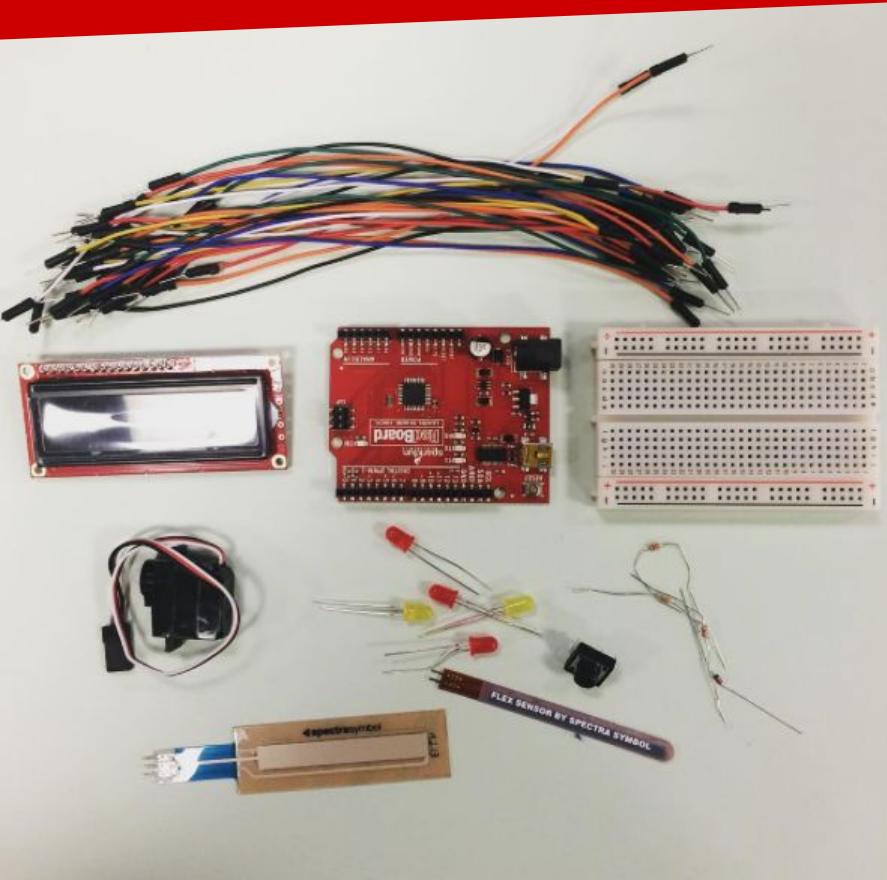
Go to: www.arduino.cc



Download the Arduino IDE

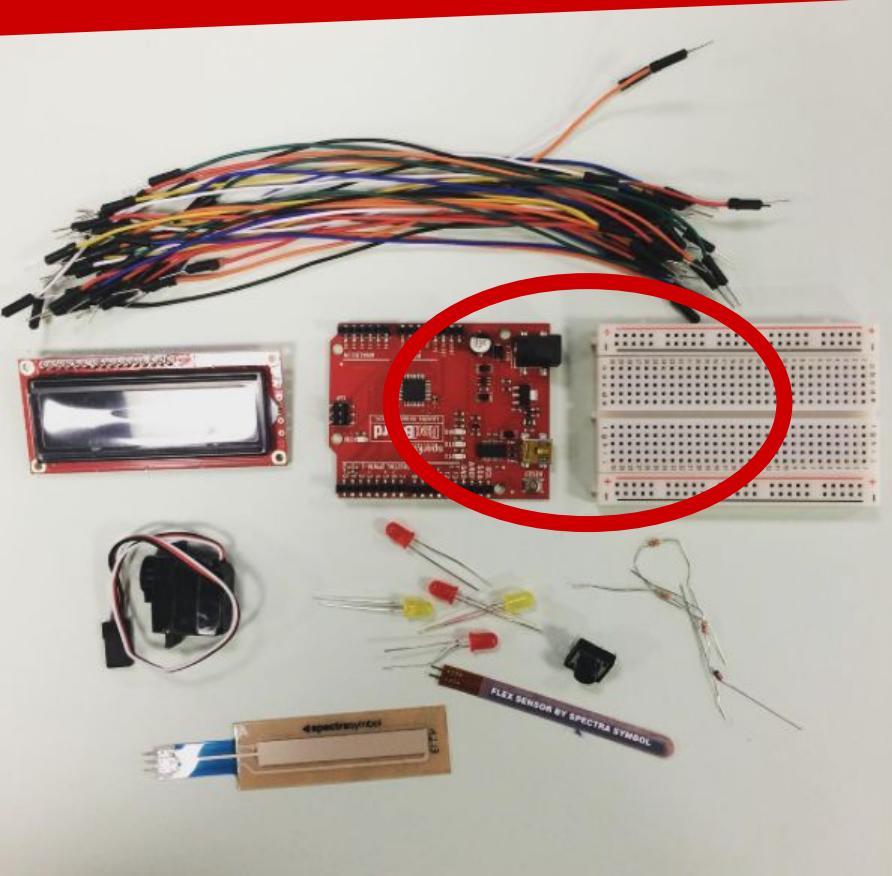


Materials



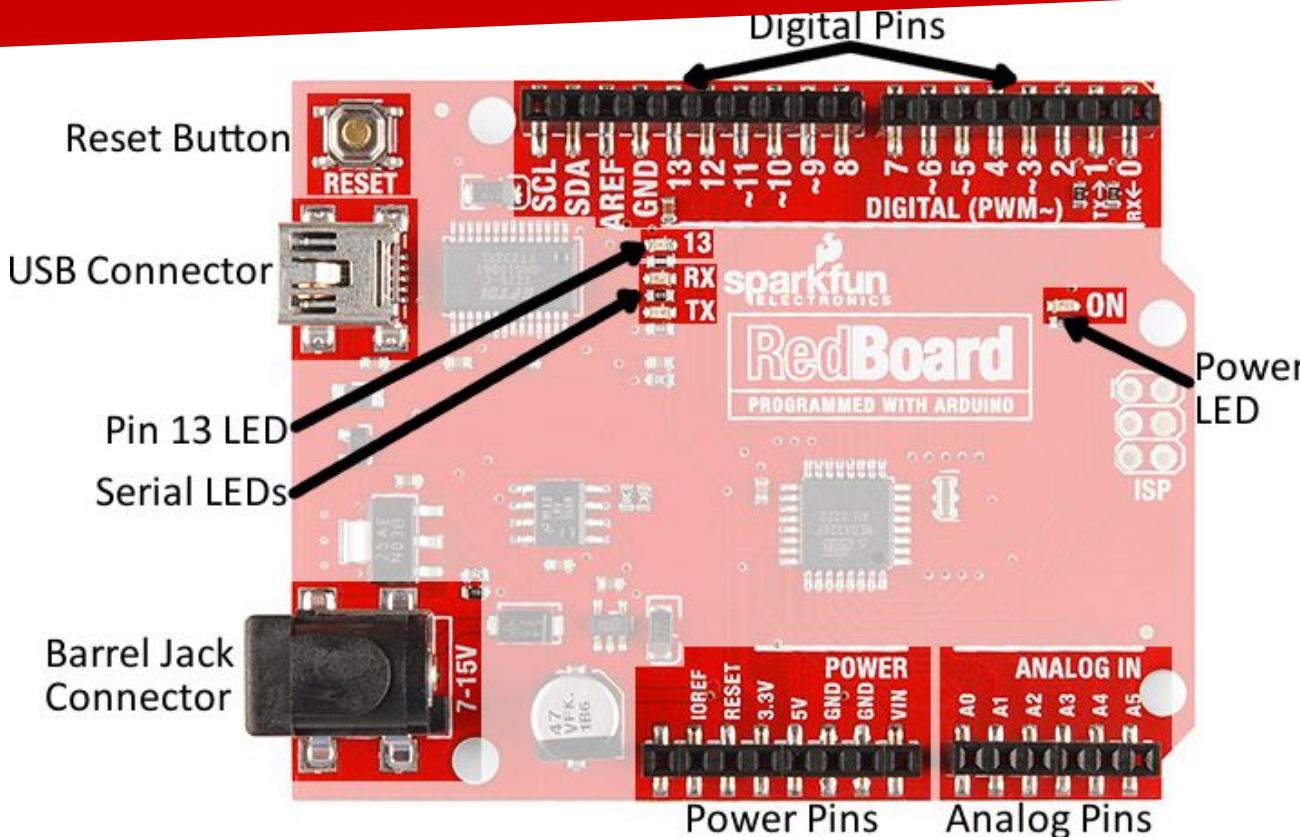
- Arduino
- Breadboard
- Components
- Jumper wires

Materials



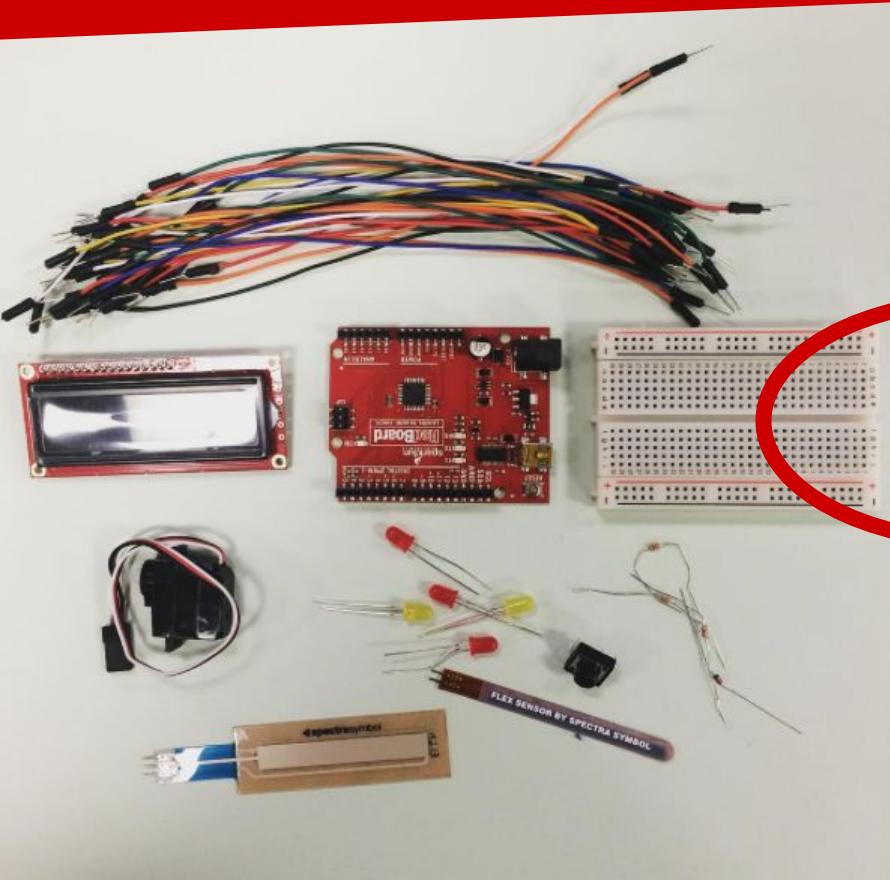
- Arduino
- Breadboard
- Components
- Jumper wires

Arduino Anatomy



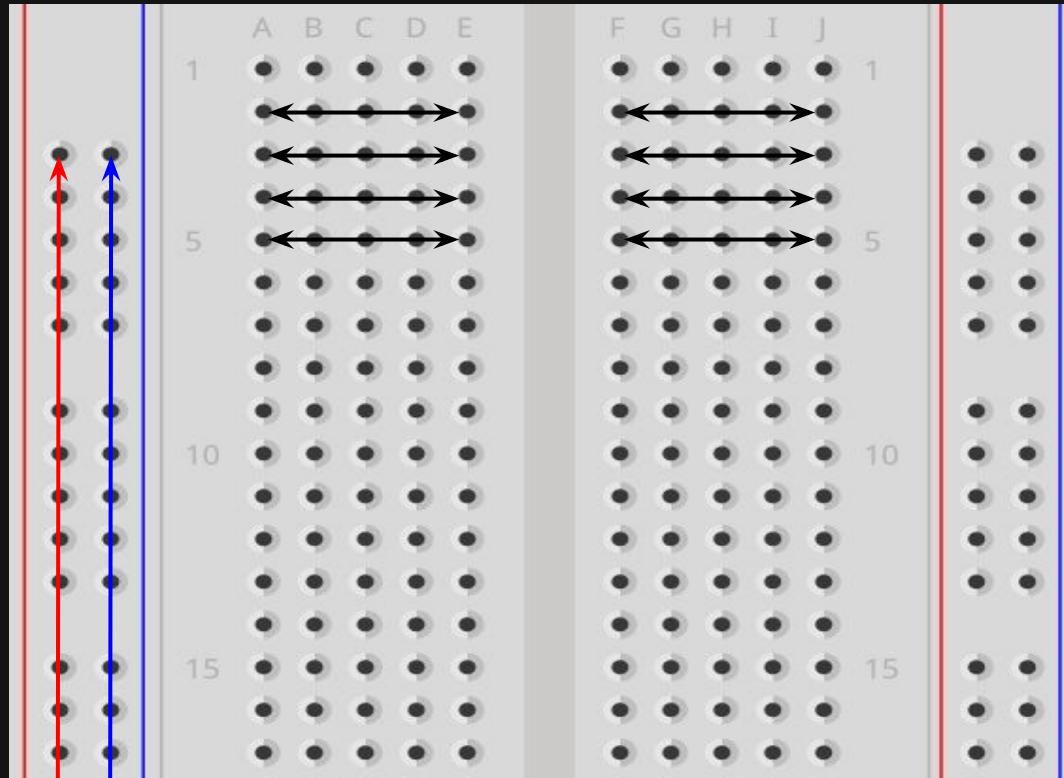
Source: <https://learn.sparkfun.com/tutorials/redboard-hookup-guide>

Materials

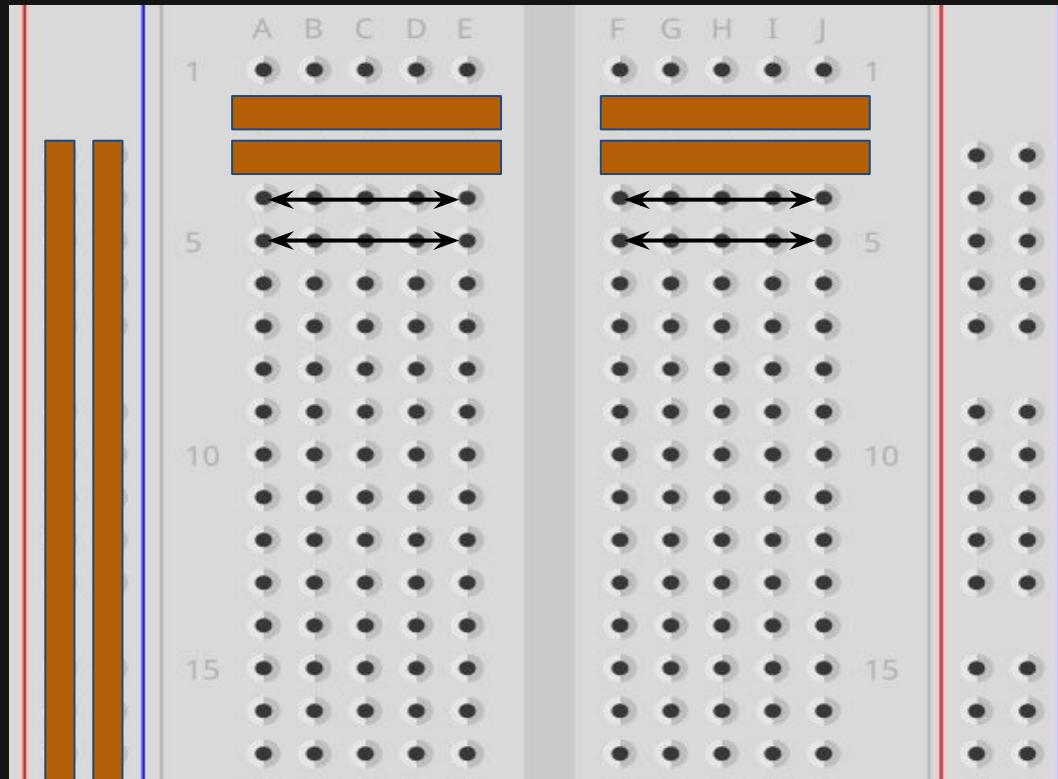


- Arduino
- Breadboard
- Components
- Jumper Wires

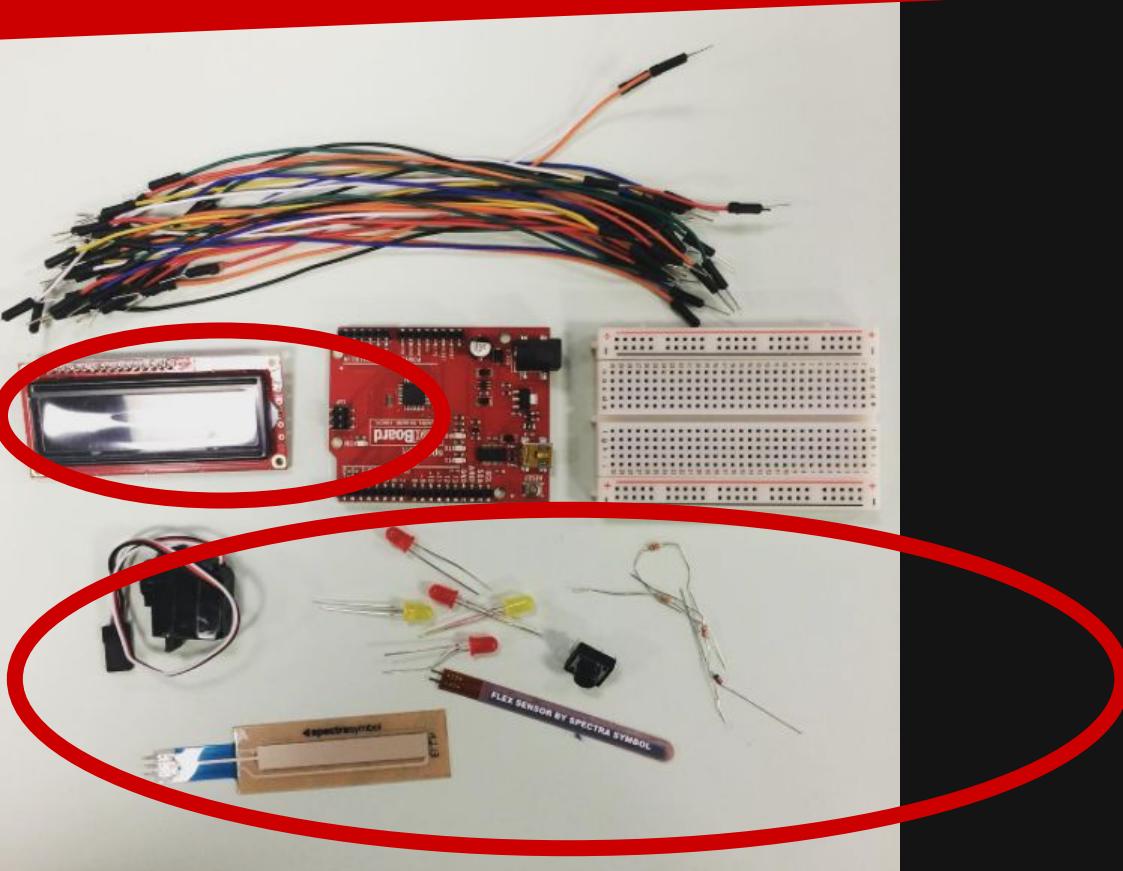
Breadboard



Breadboard



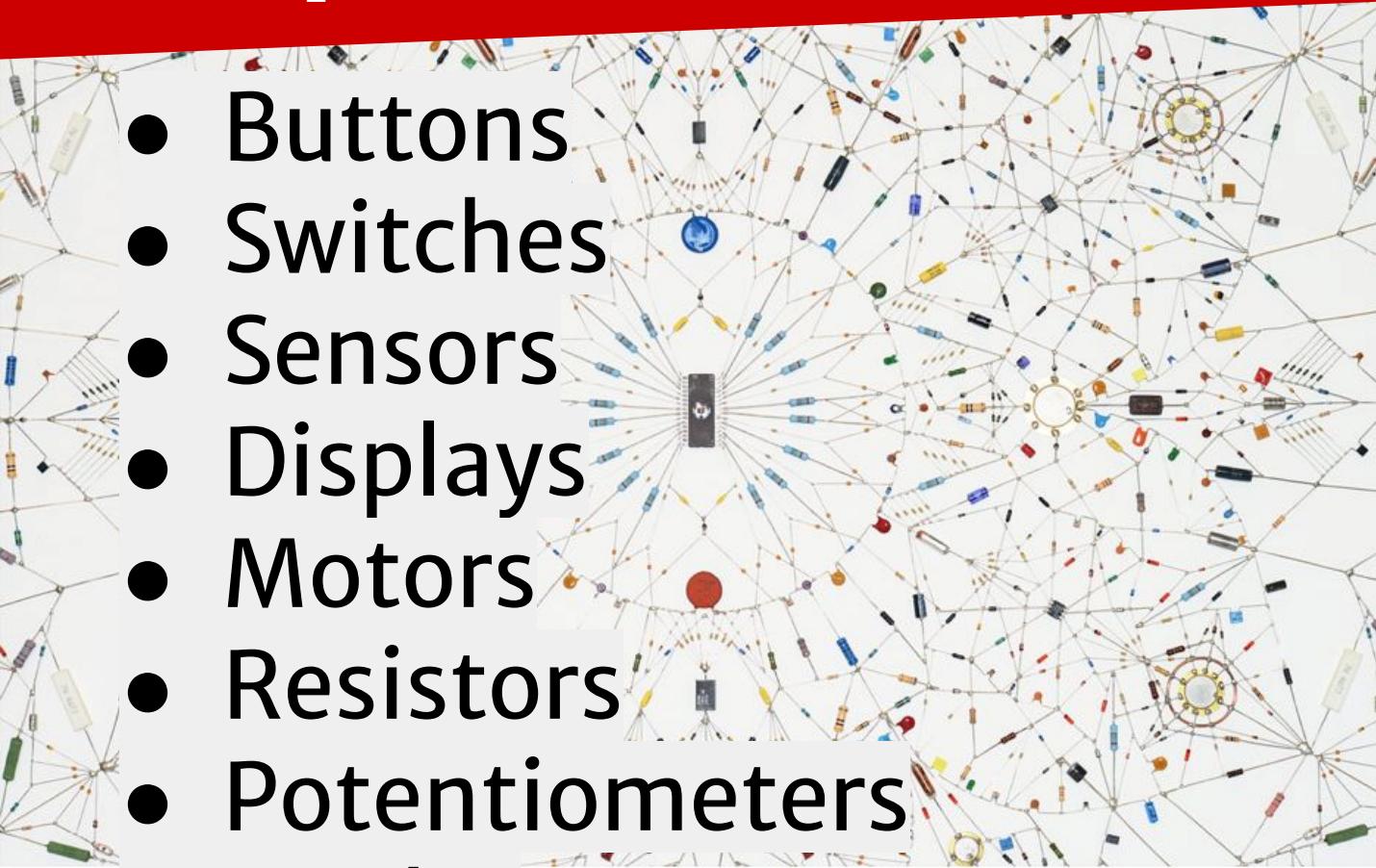
Materials



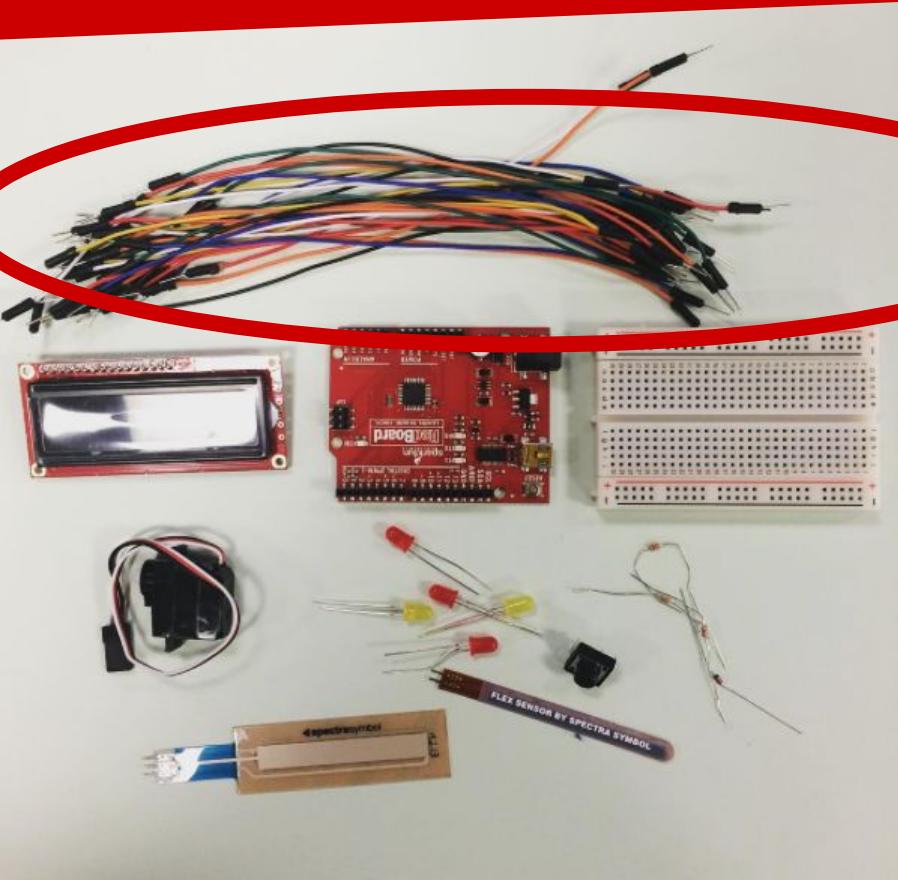
- Arduino
- Breadboard
- Components
- Jumper Wires

Components

- Buttons
- Switches
- Sensors
- Displays
- Motors
- Resistors
- Potentiometers

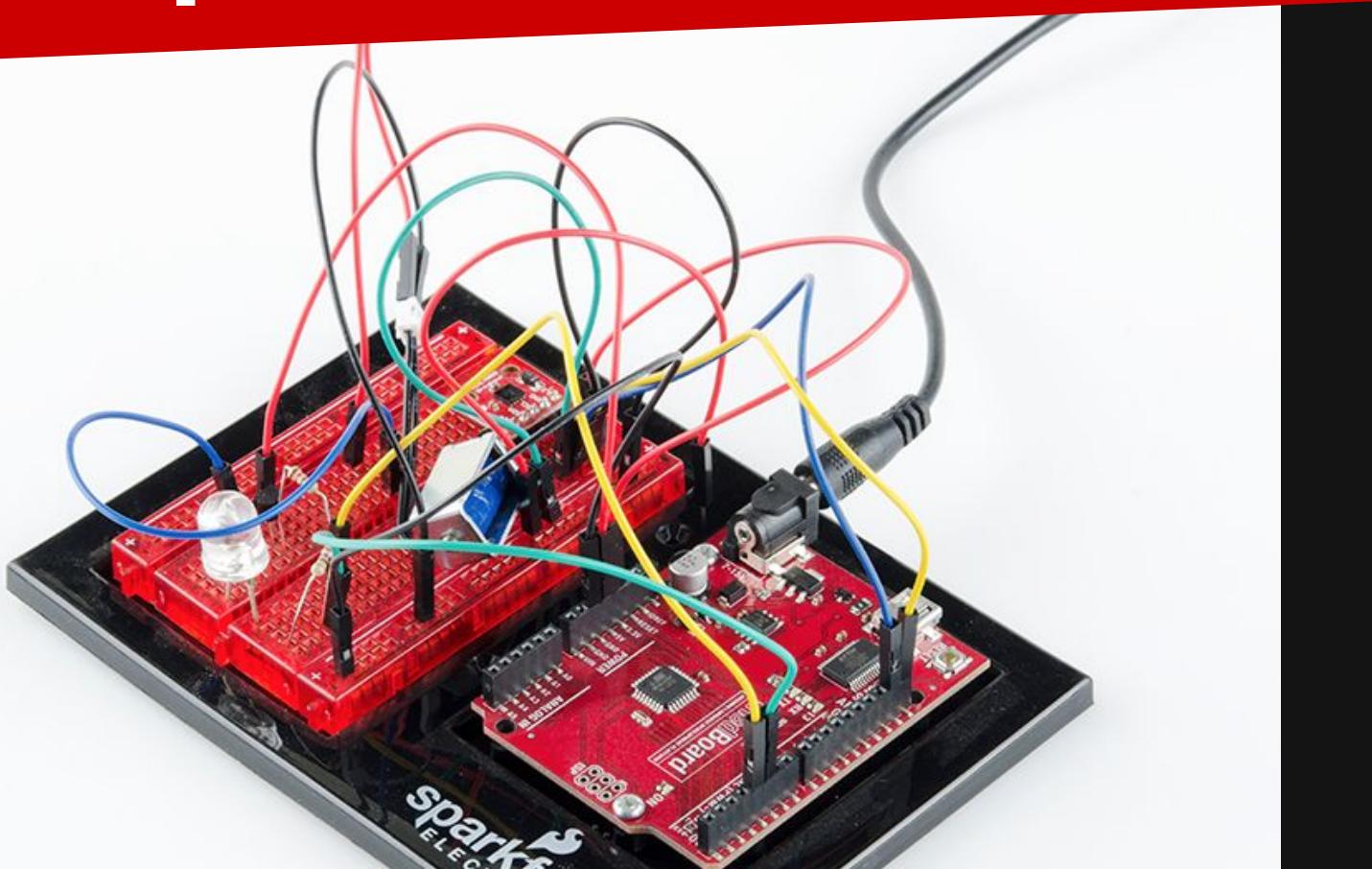


Materials

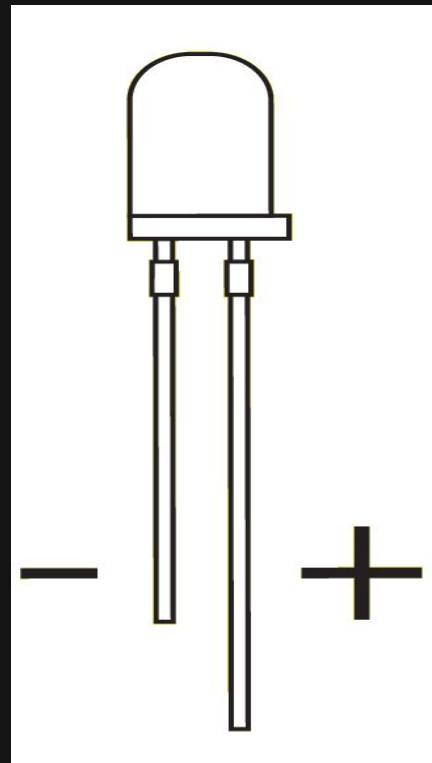
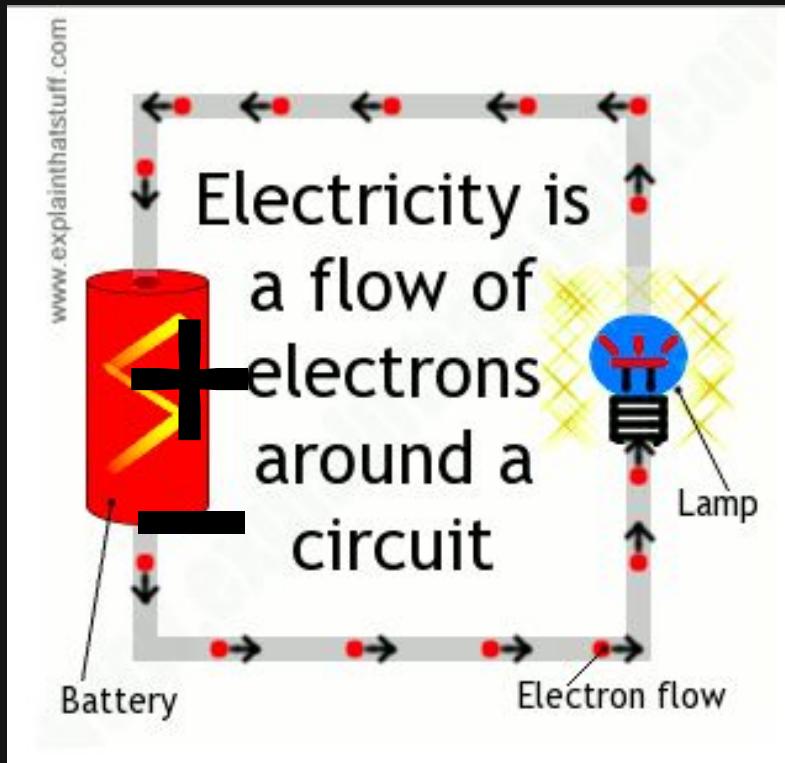


- Arduino
- Breadboard
- Components
- Jumper Wires

Jumper Wires



Circuit Basics



Today you will...

- Connect Arduino;
- Prototype a circuit;
- Upload code;
- Customize code;
- Explore!

Challenge #1

Connect an LED to pin 13

Setup

- Connect Arduino to USB
- Settings - board & port

Find...

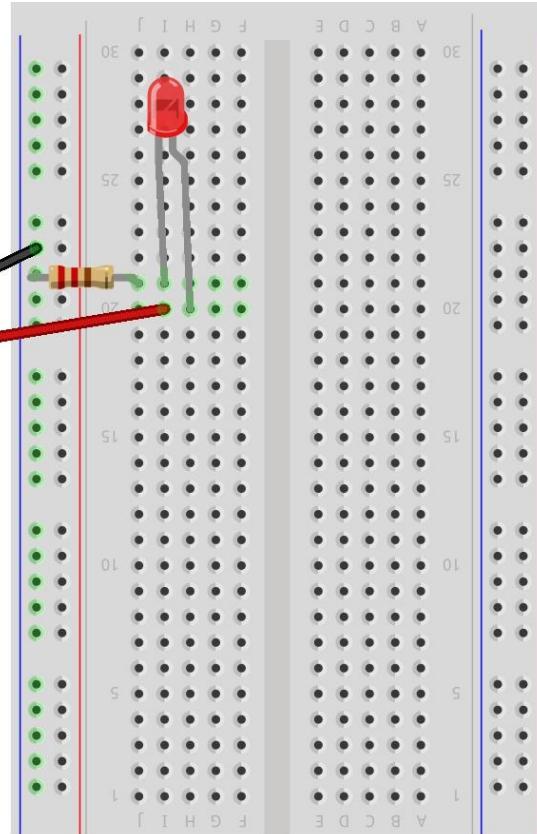
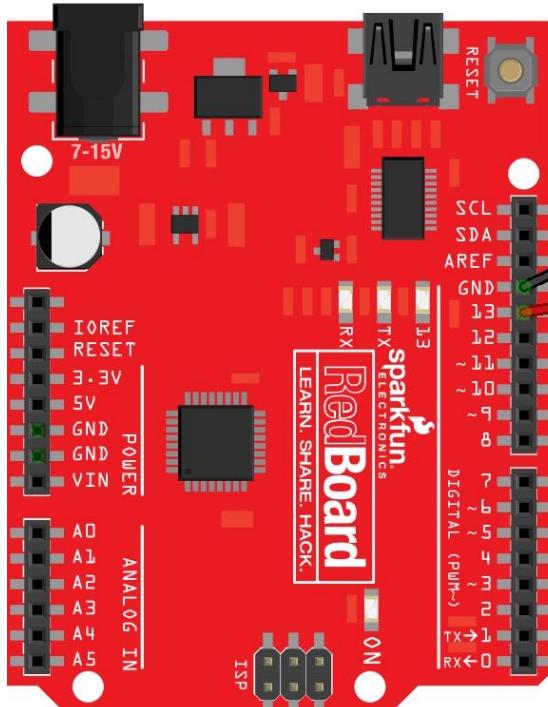


Short leg -
Long leg +



Resistor
(330ohm):
limit current

Build...

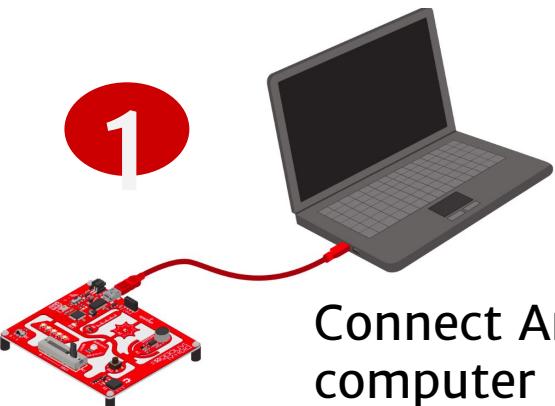


fritzing

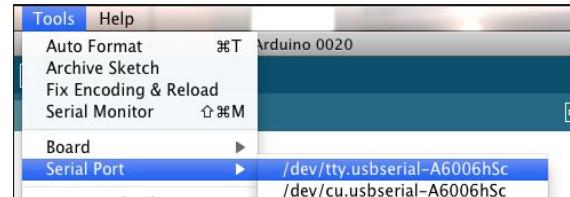
Challenge #2

Get the LED to blink.

Connect...



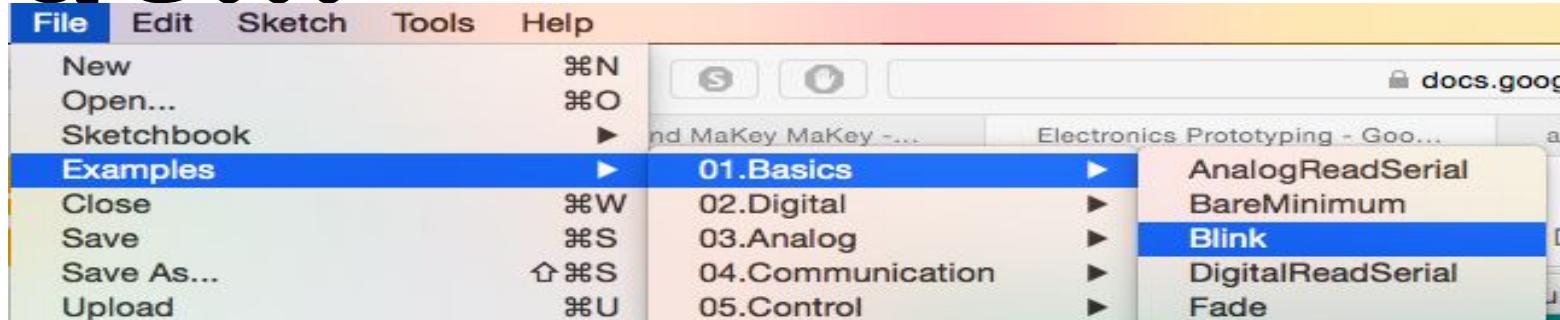
Set Board to “Arduino Uno”



Set Serial Port to “USB”

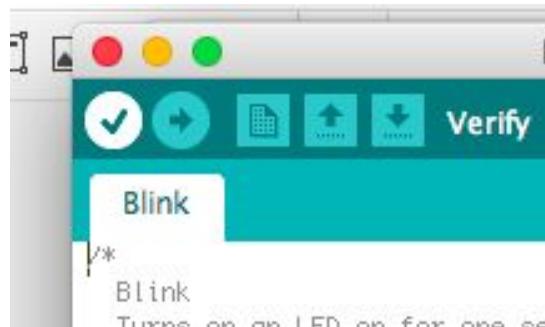
Code...

1



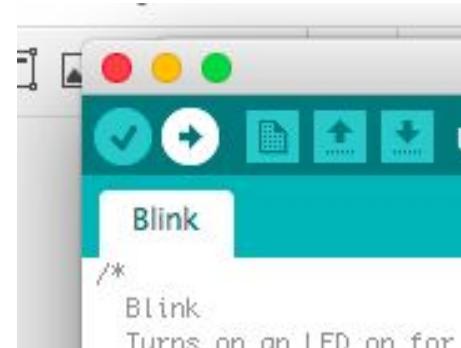
Load the example sketch “Blink.”

2



Verify the code.

3

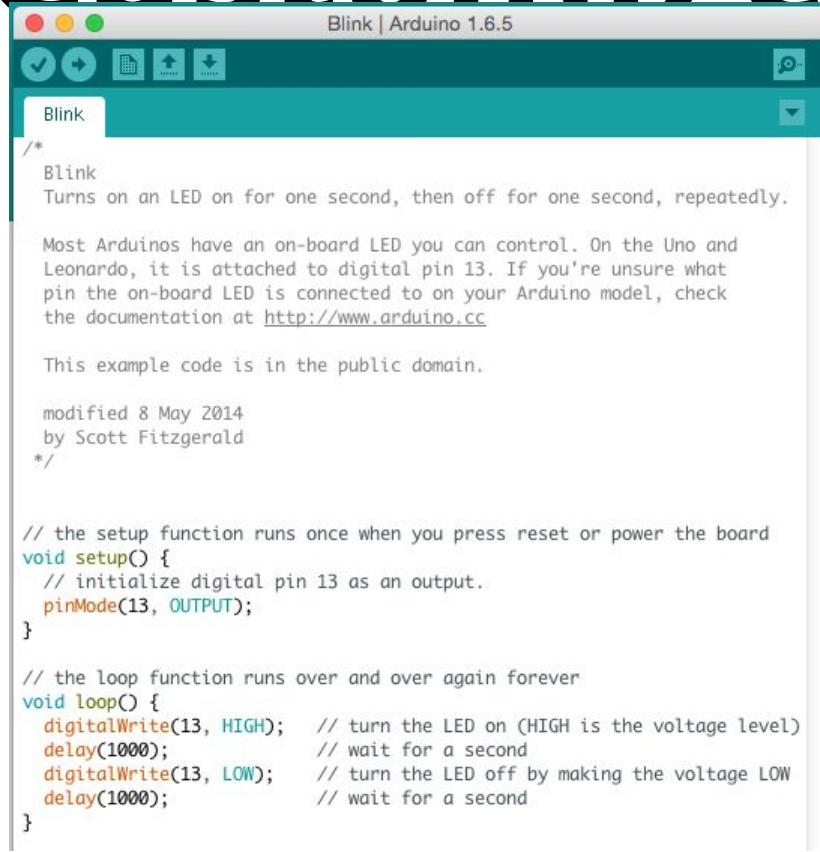


Upload to the board.

Challenge #3

Get the LED to blink
every 2.5 seconds.

Customize...



The image shows a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.6.5". The main window displays the "Blink" example sketch. The code is as follows:

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://www.arduino.cc

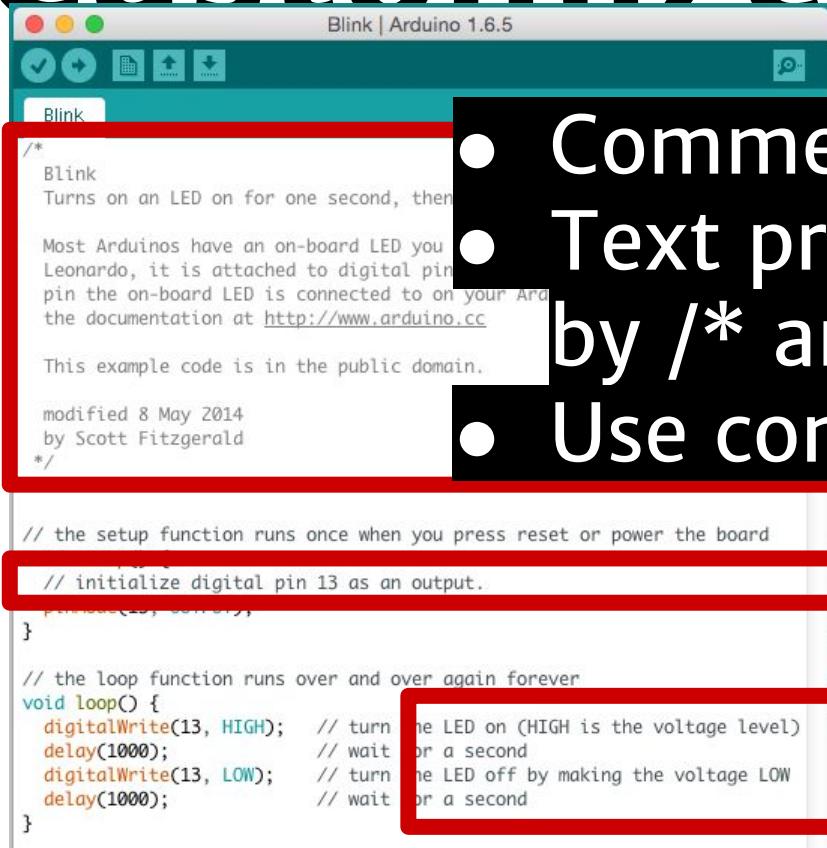
This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

Customize...



The image shows the Arduino IDE interface with the 'Blink' example sketch open. The title bar reads 'Blink | Arduino 1.6.5'. The code editor contains the standard Blink sketch. A red box highlights the explanatory text at the top of the code, which describes the purpose of the sketch and provides a link to the official documentation. Another red box highlights the first two lines of the setup function, which initializes pin 13 as an output. A third red box highlights the entire loop function, showing the code for turning the LED on and off.

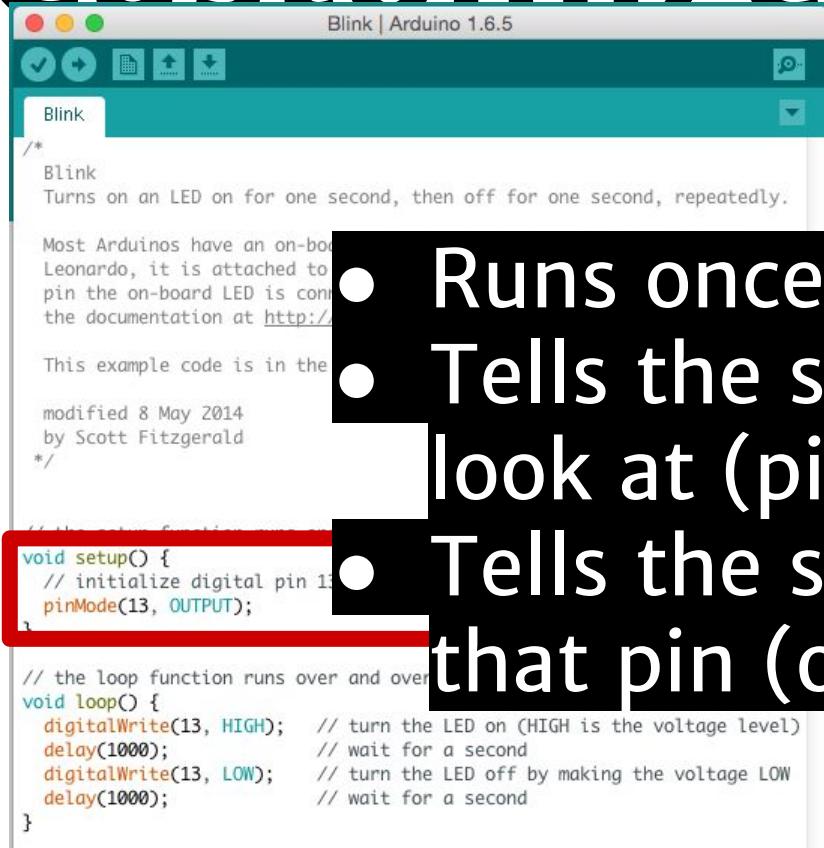
```
/*
Blink
Turns on an LED on for one second, then
turns it off for one second, repeating
this process forever. Most Arduinos have an on-board LED you can
use for this example. On the Leonardo, it is attached to digital pin 13. The
pin the on-board LED is connected to on your Arduino depends on your
board; see the documentation at http://www.arduino.cc
This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
*/
// the setup function runs once when you press reset or power the board
// initialize digital pin 13 as an output.
void setup() {
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

- Comments help explain code
- Text preceded by // or bounded by /* and */ are comments
- Use comments when you code

Customize...



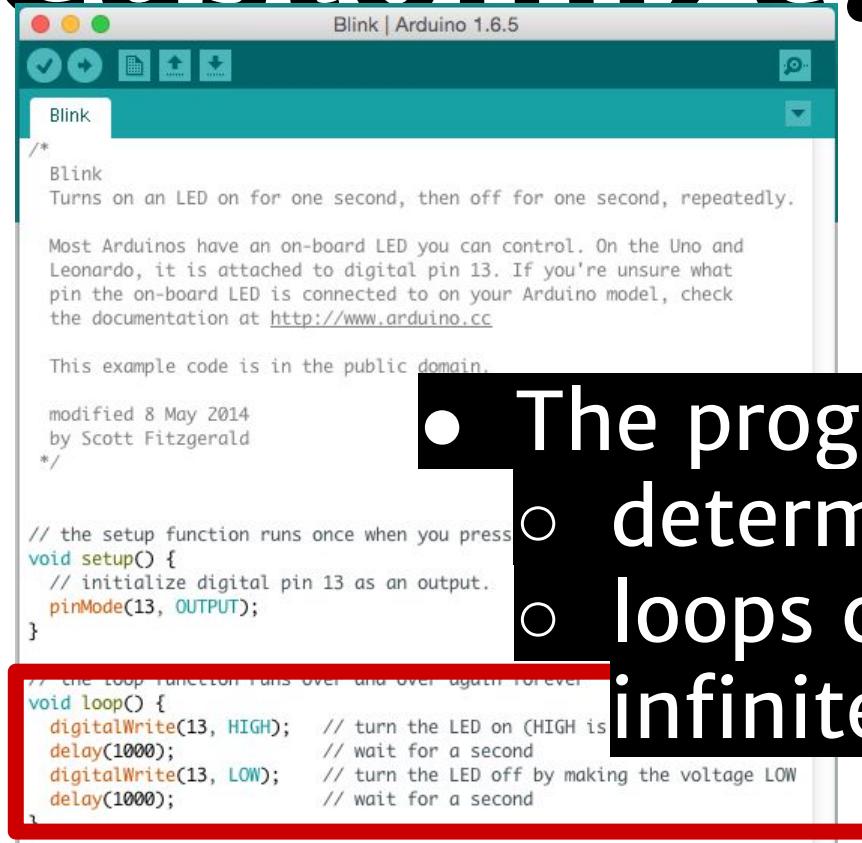
The image shows the Arduino IDE interface with the "Blink" example sketch open. The title bar reads "Blink | Arduino 1.6.5". The code editor displays the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED attached to pin 13. On the Leonardo, it is attached to pin 13. On the Uno and Duemilanove, the on-board LED is connected to pin 13. For the Uno and Duemilanove, the documentation at http://www.arduino.cc/en/Guide/ArduinoBlinks says that pin 13 is the "Digital Pin 13". This example code is in the public domain.
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 */
// the setup function runs once
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over
void loop() {
    digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
    delay(1000);              // wait for a second
    digitalWrite(13, LOW);     // turn the LED off by making the voltage LOW
    delay(1000);              // wait for a second
}
```

- Runs once
- Tells the software which pin to look at (pin 13)
- Tells the software how to treat that pin (output)

Customize...



The image shows a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.6.5". The main area displays the "Blink" sketch code. The code includes a detailed comment at the top explaining the purpose of the sketch and how to find documentation. It also includes a note about the public domain status and authorship. The setup and loop functions are shown, with the loop function highlighted by a red rectangle. The code uses standard C-like syntax for initializing pin 13 as an output and then toggling it between HIGH and LOW with a one-second delay.

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the Uno and
 * Leonardo, it is attached to digital pin 13. If you're unsure what
 * pin the on-board LED is connected to on your Arduino model, check
 * the documentation at http://www.arduino.cc
 *
 * This example code is in the public domain
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 */
// the setup function runs once when you press
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH);    // turn the LED on (HIGH is
    delay(1000);              // wait for a second
    digitalWrite(13, LOW);     // turn the LED off by making the voltage LOW
    delay(1000);              // wait for a second
}
```

- The program
 - determines behaviours
 - loops consecutively and infinitely

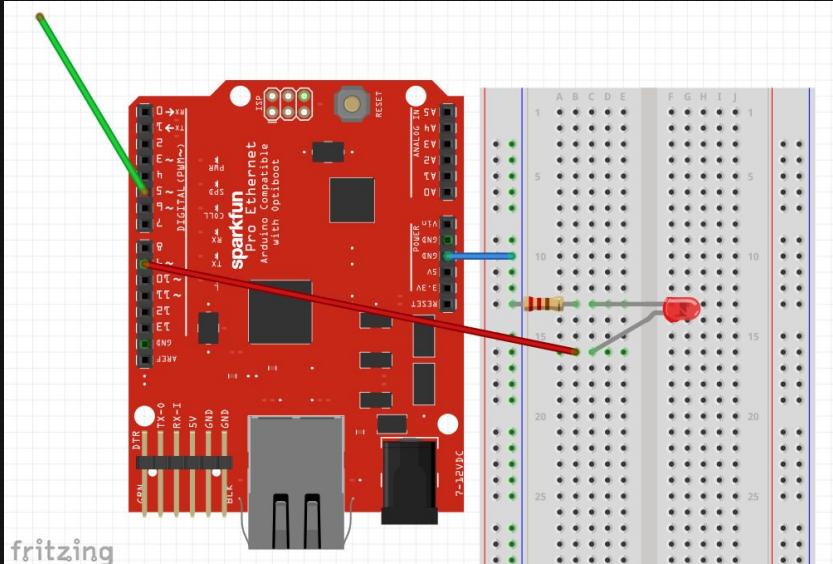
EM Field Detector

What? Why?

- What is EMF?
- What does an EMF detector do?
- Arduino -> Detector

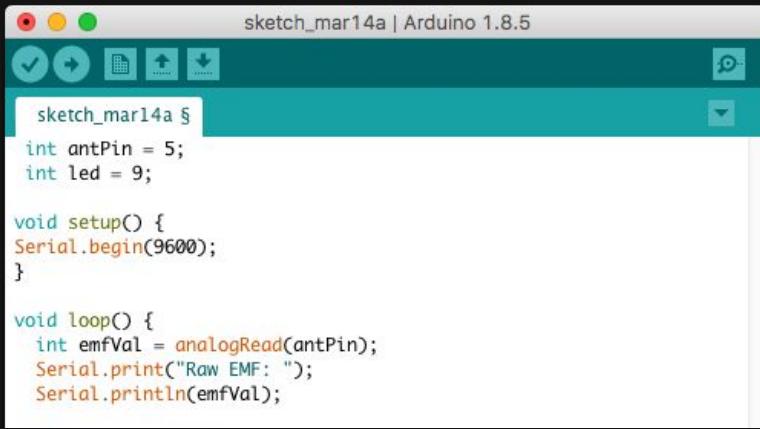
Let's Get Started!

The Circuit



1. Connect an LED to pin 9
2. Setup antenna or probe on pin 5
 - a. Today we're going to use one of our jumper wires
 - b. You can use a solid core wire, or any other conductive material
 - c. The material you use for an antenna
 - d. The length and shape of an antenna
 - e. All affect the sensitivity of your device, and the kind of readings you get
3. Good place to conduct experiments and more research

The Code



A screenshot of the Arduino IDE interface. The title bar says "sketch_mar14a | Arduino 1.8.5". The code editor window contains the following code:

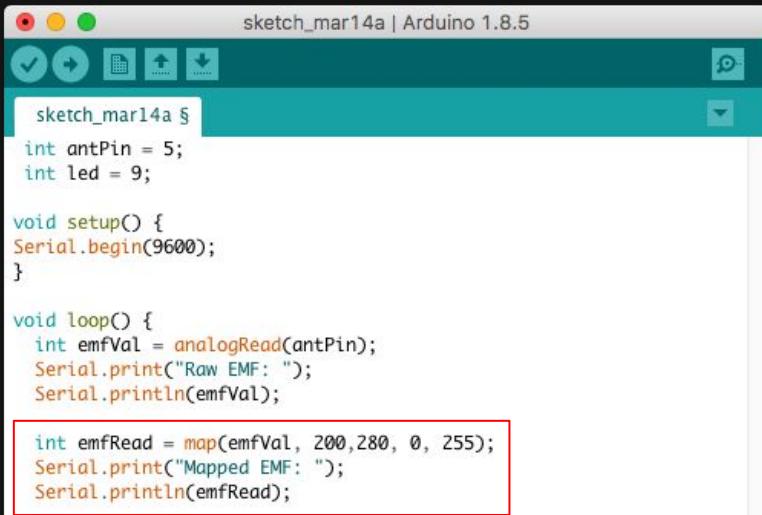
```
int antPin = 5;
int led = 9;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int emfVal = analogRead(antPin);
  Serial.print("Raw EMF: ");
  Serial.println(emfVal);
```

1. Create global variables
2. Serial for output
3. Looping
4. Upload and experiment
 - a. Find the highest and lowest values
 - b. We will need this for the next part

The Code: Part 2



The screenshot shows the Arduino IDE interface with the title bar "sketch_mar14a | Arduino 1.8.5". The code editor contains the following code:

```
int antPin = 5;
int led = 9;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int emfVal = analogRead(antPin);
  Serial.print("Raw EMF: ");
  Serial.println(emfVal);

  int emfRead = map(emfVal, 200, 280, 0, 255);
  Serial.print("Mapped EMF: ");
  Serial.println(emfRead);
}
```

A red rectangular box highlights the mapping code in the loop function:

```
int emfRead = map(emfVal, 200, 280, 0, 255);
Serial.print("Mapped EMF: ");
Serial.println(emfRead);
```

1. Use those high and low limits
 - a. I see upper 300 and lower 200
 - b. What do you see?
2. Mapping
 - a. What is it?
 - b. Why?
 - c. `int emfRead = map(emfVal, 200, 280, 0, 255);`
3. Output to Serial again
 - a. Upload and see the values

The Code: Part 3



The screenshot shows the Arduino IDE interface with the title bar "sketch_mar14a | Arduino 1.8.5". The code editor contains the following Arduino sketch:

```
int antPin = 5;
int led = 9;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int emfVal = analogRead(antPin);
  Serial.print("Raw EMF: ");
  Serial.println(emfVal);

  int emfRead = map(emfVal, 200,280, 0, 255);
  Serial.print("Mapped EMF: ");
  Serial.println(emfRead);

  if(emfVal>1){
    analogWrite(led, emfRead);
  }
  else{
    analogWrite(led,0);
  }
}
```

A red rectangular box highlights the conditional statement starting with "if(emfVal>1)".

1. LED as interface
 - a. Glanceable
 - b. Turn on the LED when EMF is present
2. If statement

```
if(emfVal>1){

  analogWrite(led,emfRead);

}

else{

analogWrite(led,0);

}
```

Experiment!

Share?

Next Steps



let's make



Explore

Publish

Login | Sign Up

share what you make

Featured:



Intel IoT



Life Hacks



Beauty



Woodworking

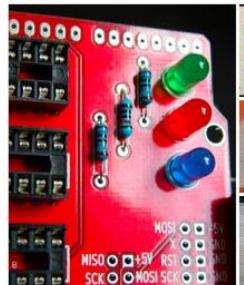
Let's Make

arduino



Arduino

by Instructables Guides in Arduino



Arduino

by playmobilby in Arduino



Arduino

by anouskadg in Arduino



ARDUINO

XYLOPHONE

instructables.com

SparkFun Electronics

Need Help? \$ USD

sparkfun

SHOP LEARN AVC FORUM DATA

LOG IN REGISTER

START A PROJECT PRODUCTS BLOG TUTORIALS VIDEOS WISH LISTS DISTRIBUTORS SUPPORT

arduino

HOME / SEARCH RESULTS FOR ARDUINO

6,064 results for arduino

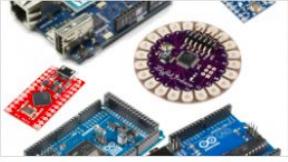
Confused about Arduino? Check out our [Arduino Buying Guide!](#)

PRODUCTS 802 TUTORIALS 364 BLOG 428 RESOURCES 43 CLASSES 0 ARTICLES 22 WISH LISTS 4403

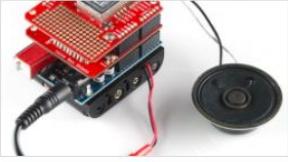
Tutorials 268 found



[What is an Arduino?](#)
FEBRUARY 26, 2013
What is this 'Arduino' thing anyway?



[Arduino Comparison Guide](#)
NOVEMBER 11, 2013
Examining the diverse world of Arduino boards.



[Arduino Shields](#)
AUGUST 6, 2013
All things Arduino Shields. What they are and how to assemble them.



[Installing an Arduino Library](#)
JANUARY 11, 2013
How do I install a custom Arduino library? It's easy!

learn.sparkfun.com

S 1

learn.adafruit.com

Sign In 0 Items

adafruit

SHOP BLOG LEARN FORUMS VIDEOS

LEARN HOME

LEARN ARDUINO

32 GUIDES | 229 PAGES | 2 FEATURED | 6 POPULAR

Set up and Blink - MATLAB and Simulink with Arduino

How to use MATLAB and Simulink to communicate with your Arduino

Color Balancing Video Camera Light feat.

Arduino Yun Temboo Twitter Tracker

learn.adafruit.com

Adding WiFi / IoT



This is in the kit! (Advanced use)

More info: go.psu.edu/ict



<https://www.lib.ncsu.edu/techlending/makerspace>

< NC State Home RESOURCES search ncsu.edu 

NCSU LIBRARIES FIND GET HELP SERVICES LIBRARIES ABOUT

Search books, articles, journals, website  ASK US | MY ACCOUNT | HOURS | FAQ | LOG OUT | CHAT NOW 

HOME / TECHNOLOGY LENDING / MAKERSPACE

Makerspace

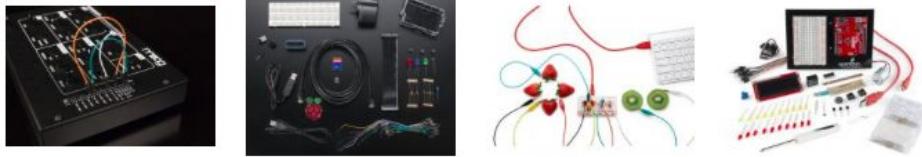


PrimeSense Carmine 1.09 3D Sensor
Combined with software, can work as a 3D scanner Capable of quick and decent...
Walk Up,
4 Hour, 7 Days

littleBits Premium Kit
The littleBits Premium Kit encourages you to learn the basics of electronics by...
Walk Up,
7 Days

Occipital Structure 3D Scanner
Checks out with an iPad Air 2 for handheld 3D scanning and data sensing On iPad,...
Walk Up,
7 Days

Intel Galileo Kit
The Intel Galileo Gen 2 board is the first in a family of Arduino-certified development...
Walk Up,
14 Days



Moog Werkstatt-01

Raspberry Pi

JoyLabz MaKey

SparkFun Arduino

Ask Us.